
H-811.S11 Hexapod System with Wave Generator Functionality

Contents

System Description2
Digital I/O Lines2
Wave Generator3
Basic Information3
Specific Wave Generator Details of H-811.S11 System4
Commandable Items4
Wave-Generator-Related Commands and Parameters5
Basic Operation6
Wave Generator Examples7
Macro Examples for Wave Generator Operation10
Command Descriptions for Digital I/O12
Command Descriptions for Wave Generator13
Pin Assignment of I/O Interface of Hexapod Controller24
Technical Data of H811B0018 Hexapod25

System Description

The H-811.S11 Hexapod system consists of the following items:

- H811B0018 Hexapod
- C887B0026 Hexapod controller

The H811B0018 Hexapod is based on the H-811.D1 Hexapod as it is described in the MS199E user manual. The delivered C887B0026 controller is based on the C-887.11 controller as it is described in the MS204E user manual.

The delivered Hexapod system differs from the standard Hexapod and standard controller in featuring the following items:

- Servo cycle time of 0.6 ms: The servo cycle time of the H-811.S11 Hexapod system's firmware allows to update position values in intervals of 0.6 ms.
- An interface for digital I/O lines and the related commands DIO and DIO? (p. 2)
- A firmware that provides the wave generator functionality with several related commands (p. 3, p. 5)
- Minimum incremental motion of 1 µm of the Hexapod axes (p. 25)

Digital I/O Lines

The digital I/O lines of the Hexapod controller are available on the **I/O** interface. See "Pin Assignment of I/O Interface of Hexapod Controller" (p. 24) for the lines and pinout.

You can set the states of output line 1 to output line 8 (TTL, active high) using the DIO command (p. 12), e.g. to trigger other devices. The lines can be set individually or all at once according to a bit pattern.

The states of input line 1 to input line 8 (TTL, active low) can be queried with the DIO? command (p. 12). These lines can be used to stop macros and to trigger certain actions in macros via the MEX command or the WAC command, respectively (see user manual of the C-887 Hexapod controller). For more information refer to "Controller Macros" in the user manual of the controller.

Digital I/O lines can also be used in conjunction with the wave generator (WGO command, p. 18):

- Digital input line 1 of the Hexapod controller can be used to start the controller's integrated wave generator.
- Digital output line 1 can be used to output pulses as long as a wave generator is running. The output line toggles between HIGH and LOW synchronized with the servo cycles.

See "Trigger Output Synchronized with Wave Generator" (p. 9) and "Wave Generator Started by Trigger Input" (p. 10) for application examples.

INFORMATION

To use the digital I/O lines to stop the wave generator, you can use a macro that performs the following steps:

- Querying the state of the digital input line 1 with DIO?
- Setting the condition that, if the digital input line 1 is set high, a WGO command is sent to switch off the wave generator

Wave Generator

Basic Information

Each axis can be controlled by a "wave generator" which outputs user-specified patterns, so-called "waveforms".

The wave generator functionality helps to avoid delays due to communication issues: A waveform stored to the controller's volatile memory ensures that the time intervals are observed.

The waveforms to be output are stored in "wave tables" in the controller's volatile memory—one waveform per wave table.

Waveforms can be created based on predefined "curve" shapes. This can be a sinusoidal or an arbitrary user-defined curve shape.

The Hexapod controller has 100 wave tables for creating and (temporarily) storing arbitrary waveforms (identifiers are 1 to 100). A wave table can be filled with up to 1,000,000 data points.

A macro can be used to fill the wave tables, see p. 10 for an example and the MS204E user manual for macro handling.

The assignment of wave generators and axes to each other is fixed: the wave generators 1 to 6 are assigned to the Hexapod axes as given in the table below.

Axis	Wave generator ID
X	1
Y	2
Z	3
U	4
V	5
W	6

The available wave tables can be flexibly assigned to the wave generators and hence to the axes using the WSL command (p. 20). A wave table can be used by multiple wave generators at the same time.

A certain amount of the controller's memory space is reserved for the waveform data (ask with the SPA? command, parameter ID 0x13000004). The Hexapod controller provides 1,000,000 data points for waveform definition. This memory space is (temporarily) allocated to the individual wave tables during the waveform definition.

The interval between two data points equals the product of servo cycle time, 0.6 ms, and the wave table rate value set by the WTR command (p. 21). A maximum wave table rate value of 1,000 is possible, resulting in a duration of 0.6 s for output of a single waveform point. At present, the same wave table rate value must be set for all wave generators.

When the wave generator is started it outputs the waveform repeatedly until it is stopped. Using the WGC command (p. 17) you can set how often a waveform is output.

Programmable digital input line 1 and output line 1 facilitate synchronization of external events. For start mode settings see WGO command description (p. 18).

After you sent the waveform definition to the wave table (with WAV), it is always a good idea to check it by reading back the waveform sequence from the controller before actually outputting it. This can be done by the GWD? command (p. 13).

In "Wave Generator Examples" (p. 7) and "Macro Examples for Wave Generator Operation" (p. 10) you will learn how to use the wave generator, and "Wave-Generator-Related Commands and Parameters" (p. 5) gives an overview.

Specific Wave Generator Details of H-811.S11 System

- When the wave generator output is active, no move command and no macro containing a move command for any axis is allowed.
- Note that when you address one wave generator using a WTR or WGC command, all other wave generators are addressed simultaneously.
- Since the motion of the moving platform always involves all axes of the Hexapod, the WGO command always starts the output of all wave generators, irrespective of the wave generator given in the command. Wave generators which are not connected to a wave table will output the last valid target position.
- Since the controller card can be working to full capacity when the wave generator is running, we recommend not to use the data recorder simultaneously.
- The frequency of the sinusoidal motion that can be performed by a Hexapod depends on the following items:
 - The axis to be moved
 - Current pivot point coordinates
 - Amplitude of the sinusoidal motion

For example, the following table contains recommended maximum frequencies for the V axis, for pivot point coordinates (0/0/0):

Amplitude (°pp)	Frequency (Hz)
4	3.3
2	6.5
1	12
0.5	18

- Motion which is started by the wave generator can be performed with a velocity which is higher than set by VLS, but with a maximum velocity of 25 mm/s for X,Y,Z.

Commandable Items

Item	Number	Identifier	Description
Wave generator	6	1 to 6	Each wave generator is dedicated to one axis, for axis assignment refer to "Basic Information" on p. 3.
Wave table	100	1 to 100	The wave tables contain the data points of the waveform to be output. The controller has 100 wave tables with max. 1,000,000 data points per table.

Wave-Generator-Related Commands and Parameters

Command	Description	Notes
GWD? (p. 13)	Get Wave Table Data	Should be used to check the waveform before the wave generator output is started.
WAV (p. 14)	Set Waveform Definition	A waveform must be defined before the wave generator output can be started.
WAV? (p. 16)	Get Waveform Definition	Reads the current wave table length.
WGC (p. 17)	Set Number Of Wave Generator Cycles	If WGC is not used, the wave generator must be stopped with WGO, #24 or STP.
WGC? (p. 18)	Get Number Of Wave Generator Cycles	
WGO (p. 18)	Set Wave Generator Start/Stop Mode	The WGO command starts the wave generator output. It provides several start modes, e.g. "Start wave generator output triggered by external signal".
WGO? (p. 20)	Get Wave Generator Start/Stop Mode	Gets the start modes while the wave generator is active
WSL (p. 20)	Set Connection Of Wave table To Wave Generator	Must be set before the wave generator can be started.
WSL? (p. 21)	Get Connection Of Wave table To Wave Generator	
WTR (p. 21)	Set Wave Generator Table Rate	Sets the value of the wave generator table rate in volatile memory; determines the interpolation type for wave table rate values > 1.
WTR? (p. 23)	Get Wave Generator Table Rate	Gets the value of the wave generator table rate from volatile memory and the current interpolation type.
#9 (p. 13)	Get Wave Generator Status	Gets the current activation status of the wave generator, but not the current start modes (use WGO? instead)

For detailed command descriptions see "Command Descriptions for Wave Generator" (p. 13).

For the identifiers of the items which can be addressed with the commands see "Commandable Items" (p. 4).

Parameter ID	Command Level	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description
0x13000004	3	System	1	INT	Max Wave Points

This parameter can be queried using the SPA? command, as it is described in the user manual of the C-887 Hexapod controller.

Basic Operation

1. Define the waveform segment-by-segment using the WAV command. The waveform will be written to the selected wave table.
2. Connect the wave generator to the wave table using the WSL command.
3. Optional: Set the number of output cycles for the wave generator using the WGC command.
4. Start the wave generator output and hence the motion of the axis using the WGO command. You can choose several start modes (e.g. start by external trigger, see the description of the WGO command (p. 18) for more information).
5. Stop the wave generator output with WGO or #24 or STP. If the number of output cycles was limited with WGC, the output will stop automatically after the given number of cycles.

Simple examples for your first steps (using the command entry facilities of PIMikroMove or PITerminal):

Command String to Send	Action Performed
WAV 4 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for wave table 4; see WAV description for details
WSL 1 4	Connect the wave generator 1 (X axis) to wave table 4
WGO 1 1	Start wave generator output immediately (synchronized by servo cycles) Although the command is sent for wave generator 1, all wave generators are started.
WGO 1 0	Stop wave generator output and hence the corresponding Hexapod motion Although the command is sent for wave generator 1, all wave generators are stopped..

Command String to Send	Action Performed
WAV 1 X PNT 1 1 0	Set the first point in wave table 1 to target position 0; see WAV description for details
WAV 1 & sin_p 20 0.05 -0.025 20 5 10	Define a sine waveform for wave table 1 (will be concatenated to the first point 1 which was written before); see WAV description for details
WSL 3 1	Connect the wave generator 3 (Z axis) to wave table 1
WGC 3 100	Limit the number of output cycles for wave generator 3 to 100 cycles. The setting will be valid for all wave generators.
WGO 3 1	Start wave generator output immediately (synchronized by servo cycles). Although the command is sent for wave generator 3, all wave generators are started. All wave generator output automatically stops after 100 cycles.

Wave Generator Examples

Defining Wave Forms

Examples for how to define waveform segments for the wave tables, based on predefined curve shapes (each WAV command defines a waveform segment which either replaces or is appended to the waveform in the specified wave table):

Sine curves

WAV command	Comments	Waveform Segment
<pre>WAV 2 X SIN_P 2000 20 10 2000 0 1000 <Wave tableID> = 2 <AppendWave> = X <WaveType> = SIN_P <SegLength> = 2000 <Amp> = 20 <Offset> = 10 <WaveLength> = 2000 <StartPoint> = 0 <CurveCenterPoint> = 1000</pre>	The previous contents of the wave table are overwritten by the new segment, waveform offset = 10, symmetric curve	
<pre>WAV 2 X SIN_P 2000 30 0 2000 499 1000 <Wave tableID> = 2 <AppendWave> = X <WaveType> = SIN_P <SegLength> = 2000 <Amp> = 30 <Offset> = 0 <WaveLength> = 2000 <StartPoint> = 499 <CurveCenterPoint> = 1000</pre>	The previous contents of the wave table are overwritten by the new segment, symmetric curve	

WAV command	Comments	Waveform Segment
<pre>WAV 2 & SIN_P 2000 25 0 1800 100 900 <Wave tableID> = 2 <AppendWave> = & <WaveType> = SIN_P <SegLength> = 2000 <Amp> = 25 <Offset> = 0 <WaveLength> = 1800 <StartPoint> = 100 <CurveCenterPoint> = 900</pre>	The defined segment will be appended to the existing wave table contents, symmetric curve	
<pre>WAV 3 X SIN_P 4000 20 0 4000 0 3100 <Wave tableID> = 3 <AppendWave> = X <WaveType> = SIN_P <SegLength> = 4000 <Amp> = 20 <Offset> = 0 <WaveLength> = 4000 <StartPoint> = 0 <CurveCenterPoint> = 3100</pre>	The previous contents of the wave table are overwritten by the new segment, asymmetric curve	
<pre>WAV 2 X SIN_P 1000 -30 45 1000 0 500 <Wave tableID> = 2 <AppendWave> = X <WaveType> = SIN_P <SegLength> = 1000 <Amp> = -30 <Offset> = 45 <WaveLength> = 1000 <StartPoint> = 0 <CurveCenterPoint> = 500</pre>	The previous contents of the wave table are overwritten by the new segment, negative-amplitude curve, symmetric curve	

Modifying the Wave Generator Table Rate

An example for how to modify the duration of the wave generator output using the wave table rate:

Command String to Send	Action Performed
WAV 2 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for wave table 2, the segment length and hence the number of points in the wave table is 2000
WTR?	Ask for the current wave table rate (WTR value) and interpolation settings, default is wave table rate = 1 (i.e. each wave table point will be output for a duration of one servo cycle). The duration of one wave generator output cycle will be: Servo Cycle Time (in s) * WTR value * Number of Points = Output Duration (in s) 0.0006 s * 1 * 2000 = 1.2 s
WTR 1 3 1	Set the wave table rate to 3, tripling the duration of one wave generator output cycle, with linear interpolation (each wave table point will now "occupy" 3 servo cycles, but with linear interpolation of the output value applied to smoothen the output). Duration of one output cycle will now be: 0.0006 s * 3 * 2000 = 3.6 s Note that WTR specifies the wave table rate for all wave generators in the system, irrespective of the wave generator ID given in the command (1 in this example).

Trigger Output Synchronized with Wave Generator

Trigger output on digital output line 1 is generated as a sequence of pulses. The trigger output starts when the first point of the waveform is output by the wave generator and runs as long as a wave generator is running. The pulse width is one servo cycle, and the time between pulses is also one servo cycle, i.e. the output line toggles between HIGH and LOW synchronized with the servo cycles.

An example for how to generate trigger pulses synchronized with the wave generator:

Command String to Send	Action Performed
WAV 2 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for wave table 2, the segment length and hence the number of points in the wave table is 2000
WSL 1 2	Connect wave generator 1 (X axis) to wave table 2
WTR 1 3 1	Set wave table rate value for wave generator 1 (and thus for all wave generators) to 3. Use straight line as interpolation type
WGO 1 9	Start mode 1 (decimal format) and start option 8 (decimal format) are combined in one command: Start wave generator output immediately (synchronized by servo cycle) and start also the output of pulses on digital output line 1 (synchronized by servo cycle). Although the command is sent for wave generator 1, all wave generators are started.

Command String to Send	Action Performed
WGO 1 0	Stop wave generator output and hence also the trigger output and any Hexapod motion due to a wave generator output. You can also use #24 or STP. Although the command is sent for wave generator 1, all wave generators are stopped.

Wave Generator Started by Trigger Input

Using digital input line 1 of the Hexapod controller, it is possible to apply start signals for the wave generator output.

An example for how to start the wave generator by external trigger signals:

Command String to Send	Action Performed
WAV 2 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for wave table 2, the segment length and hence the number of points in the wave table is 2000
WSL 1 2	Connect wave generator 1 (X axis) to wave table 2
WGO 1 2	Start output of wave generator 1 triggered by external signal. To provide an external signal, the digital input line 1 can be used. Although the command is sent for wave generator 1, all wave generators will be started by the trigger.
WGO 1 0	Stop output of wave generator 1 (any further trigger pulses will be ignored) and any Hexapod motion due to a wave generator output. You can also use #24 or STP. Although the command is sent for wave generator 1, all wave generators are stopped.

Macro Examples for Wave Generator Operation

Macro “uvdata”

Macro “uvdata” defines waveforms for wave tables 1 and 2. The waveforms are intended to be used for axes U and V, see macro “start”.

INFORMATION

For clarity, only a single position is defined per command line in the example given below. However, to allow for quicker processing of the macro we recommend to define several positions per command line: 6 positions with 6 decimal places per command line are possible.

Command lines:

```
WAV 1 X PNT 1 1 0
WAV 1 & PNT 1 1 5.65462531935645E-06
WAV 1 & PNT 1 1 3.09100495175123E-05
[...]
WAV 1 & PNT 1 1 0.00014823366192662
WAV 2 X PNT 1 1 0
WAV 2 & PNT 1 1 -4.74244418938712E-06
WAV 2 & PNT 1 1 -3.02735241470474E-05
[...]
WAV 2 & PNT 1 1 -0.000257643502699756
```

Macro “start”

Macro “start” performs the following steps:

1. Macro “start” starts macro “uvdata”, which is used for defining waveforms for wave tables 1 and 2.
2. Connects wave tables 1 and 2 with wave generator 4 for U axis and wave generator 5 for V axis using the WSL command
3. Sets the output duration of one wave table point to 5 x 0.6 ms using the WTR command
4. Defines that the waveforms are output a 100 times using the WGC command
5. Starts all wave generators using the WGO command

Command lines:

```
MAC START uvdata
WSL 4 1 5 2
WTR 1 5 1
WGC 1 100
WGO 1 1
```

INFORMATION

If macro “start” is to be used as start-up macro, it has to start the reference move of the Hexapod before the wave generator output is started. In this case, the following lines must also be part of macro “start”:

```
FRF X
WAC FRF? X = 1
```

Command Descriptions for Digital I/O

DIO (Set Digital Output Line)

Description:	Switches the specified digital output line(s) to specified state(s).
Format:	DIO {<DIOID> <OutputOn>}
Arguments:	<DIOID> is one digital output line of the controller, see below for details. <OutputOn> is the state of the digital output line, see below for details.
Response:	none
Notes:	Using the DIO command with the H-811.S11 Hexapod system, you can activate/deactivate digital output lines 1 to 8, which are located on the I/O interface. The <DIOID> identifiers to use for the lines are 1 to 8. If <OutputOn>=1 the line is set to HIGH/ON, if <OutputOn>=0 it is set to LOW/OFF.

DIO? (Get Digital Input Lines)

Description:	Gets the states of the specified digital input lines.
Format:	DIO? [{<DIOID>}]
Arguments:	<DIOID> is the identifier of the digital input line, see below for details.
Response:	{<DIOID>=<InputOn> LF} where <InputOn> indicates the state of the digital input line, see below for details.
Notes:	With the H-811.S11 Hexapod system you can use the DIO? command to directly read digital input lines 1 to 8, which are located on the I/O interface. The <DIOID> identifiers to use for the lines are 1 to 8. If <InputOn>=1, the digital input is LOW/OFF, if <InputOn>=0, the digital input is HIGH/ON.

Command Descriptions for Wave Generator

#9 (Get Wave Generator Status)

Description:	<p>Requests the status of the wave generator(s).</p> <p>The #9 single-character command can be used to query the current activation state of the wave generators. The reply shows if a wave generator is running or not, but does not contain any information about the wave generator start mode.</p> <p>With WGO? you can ask for the current wave generator start modes while the wave generator is active (WGO settings (p. 18)).</p>
Format:	#9 (single ASCII character number 9)
Arguments:	none
Response:	<p>The answer <uint> is bit-mapped and returned as the hexadecimal sum of the following codes:</p> <p>1 = Wave Generator 1 is running, 2 = Wave Generator 2 is running, 4 = Wave Generator 3 is running, etc.</p>
Examples:	<p>0 indicates that no wave generator is running 5 indicates that wave generators 1 and 3 are running</p>

GWD? (Get Wave Table Data)

Description:	<p>Query waveform shape for given wave table.</p> <p>Depending on the waveform definition with WAV (p. 14), the wave tables may have different lengths. Due to the GCS array response format definition, it is not possible to read from tables of different lengths with one command line.</p>
Format:	<p>If the length of the wave tables differs, only tables with identical length can be read with the same command:</p> <p>GWD? <StartPoint> <NumberOfPoints> {<WaveTableID>}</p> <p>If <i>all</i> wave tables have the same length, arguments are optional as follows:</p> <p>GWD? [<StartPoint> [<NumberOfPoints> [<WaveTableID>]]]</p>
Arguments:	<p><StartPoint> is the start point in the wave table, starts with index 1</p> <p><NumberOfPoints> is the number of points to be read per table</p> <p><WaveTableID> is one wave table of the controller</p>
Response:	The wave table contents (waveform) in GCS array format (see the separate manual for the GCS array, SM 146E, and the example below)
Example:	wav 1 x pnt 1 5 1 2 3 4 5

```

gwd?
# TYPE = 1
#
# SEPARATOR = 32
# DIM = 1
# SAMPLE_RATE = 0.006000
# NDATA = 5
# NAME0 = Wave table 1
# END_HEADER
1
2
3
4
5

```

WAV (Set Waveform Definition)

Description:	<p>Define waveform of given type for given wave table.</p> <p>To allow for flexible definition, a waveform (wave table contents) can be built up by adding "segments". Each segment is defined with a separate WAV command. To add a segment, the <AppendWave> argument (see below) is used to concatenate the new segment to the existing wave table contents. (To change individual segments later, or to modify their order, the complete waveform must be recreated segment-by-segment.)</p> <p>A segment can be based on predefined "curve" shapes (see the <WaveType> argument below).</p> <p>Waveforms cannot be changed while they are being output by a wave generator. If you want to modify a waveform with WAV, first stop any wave generator output from the associated wave table.</p> <p>The waveform values are absolute values.</p>
--------------	--

	<p>The frequency of the wave generator output depends, among other factors, on the wave table length. When you create waveforms, keep in mind that the usable frequency is limited by the capability of the Hexapod mechanics.</p> <p>The duration of one output cycle for the waveform can be calculated as follows:</p> <p>Output Duration = Servo Cycle Time * WTR value * Number of Points</p> <p>where</p> <p>Servo Cycle Time = 0.6 ms</p> <p>WTR value (i.e., wave table rate) gives the number of servo cycles the output of a waveform point lasts, default is 1</p> <p>Number of Points is the length of the wave table (which is the sum of the lengths of all segments in this table)</p> <p>For more information see "Wave Generator" (p. 3).</p>
Format:	WAV <Wave tableID> <AppendWave> <WaveType> <WaveTypeParameters>
Arguments:	<p><Wave tableID> is the wave table identifier.</p> <p><AppendWave> This can be "X" or "&": "X" clears the wave table and starts writing with the first point in the table. "&" appends the defined segment to the already existing wave table contents (i.e. concatenates a segment to lengthen the waveform).</p> <p><WaveType> The type of curve used to define the segment. This can be one of: "PNT" (user-defined curve) or "SIN_P" (inverted cosine curve)</p> <p><WaveTypeParameters> stands for the parameters of the curve and can be as follows:</p> <p>For "PNT": <WaveStartPoint> <WaveLength> {<WavePoint>}</p> <p><WaveStartPoint> The index of the starting point. Must be 1.</p> <p><WaveLength> The number of curve points which will be written to the wave table (= segment length).</p> <p><WavePoint> The value of one single point.</p>

For "SIN_P":

<SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <CurveCenterPoint>

<SegLength>: The length of the wave table segment in points. Only the number of points given by <SegLength> will be written to the wave table.

<Amp>: The amplitude of the sine curve.

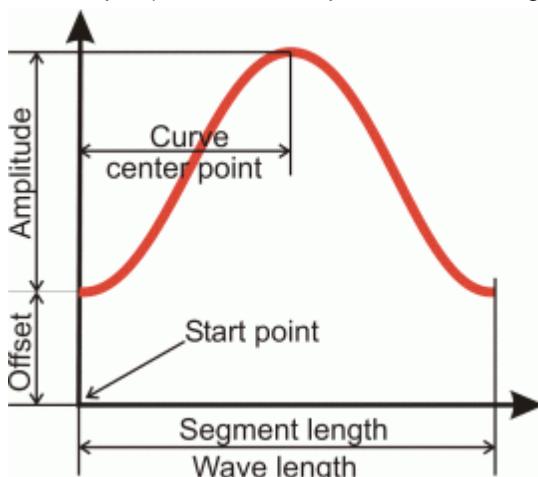
<Offset>: The offset of the sine curve.

<WaveLength>: The length of the sine curve in points.

<StartPoint>: The index of the starting point of the sine curve in the segment. Gives the phase shift. Lowest possible value is 0.

<CurveCenterPoint>: The index of the center point of the sine curve. Determines if the curve is symmetrical or not. Lowest possible value is 0.

Example (for more examples see "Defining Wave Forms", p. 7):



Response:	None
Troubleshooting:	<p>Invalid wave table identifier</p> <p>The total number of points for the waveform (which may consist of several segments) exceeds the available number of memory points.</p>

WAV? (Get Waveform Definition)

Description:	Get the value of a wave parameter for a given wave table. For more information see "Wave Generator" (p. 3).
Format:	WAV? [{<Wave tableID> <WaveParameterID>}]

Arguments:	<Wave tableID> is the wave table identifier. <WaveParameterID> is the wave parameter ID, 1 = current wave table length in number of points
Response:	{<Wave tableID> <WaveParameterID>"=<float> LF} where <float> depends on the <WaveParameterID>; gives the current number of waveform points in the wave table for <WaveParameterID> = 1
Troubleshooting:	Invalid wave table identifier

WGC (Set Number Of Wave Generator Cycles)

Description:	Sets the number of output cycles for the given wave generator (the output itself is started with WGO (p. 18)). The generator will be stopped when the number of cycles given by WGC are completed, irrespective of any further trigger pulses. With the Hexapod system, you address all wave generators simultaneously when you address any single wave generator. For more information see “Wave Generator” (p. 3).
Format:	WGC {<WaveGenID> <Cycles>}
Arguments:	<WaveGenID> is the wave generator identifier <Cycles> is the number of wave generator output cycles. If cycles = 0 then the waveform is output without period limitation until it is stopped by WGO (p. 18) or #24 or STP (see user manual of the C-887 Hexapod controller). <WaveGenID> is the wave generator identifier
Response:	None
Example:	WGC configures the wave generator 3, and thus all other wave generators simultaneously, to output its waveform twice: WGC 3 2

WGC? (Get Number Of Wave Generator Cycles)

Description:	Gets the number of output cycles set for the given wave generator. For more information see "Wave Generator" (p. 3).
Format:	WGC? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier
Response:	{<WaveGenID>}"=<Cycles> LF where <Cycles> is the number of wave generator output cycles set with WGC (p. 17).

WGO (Set Wave Generator Start/Stop Mode)

Description:	Start and stop all wave generators in the given mode. Since the motion of the moving platform always involves all axes of the Hexapod, the WGO command always starts the output of all wave generators, irrespective of the wave generator given in the command. Wave generators which are not connected to a wave table will output the last valid target position. The number of output cycles can be limited by WGC (p. 17). Using the WTR command (p. 21), you can lengthen the individual output cycles of the waveform. Keep in mind that wave generator output will continue even if the terminal or the program from which it was started is quit. The #9 single-character command (p. 13) can be used to query the current activation state of the wave generators. The reply shows if a wave generator is running or not, but does not contain any information about the wave generator start mode. With WGO? you can ask for the wave generator start modes (WGO settings) while the wave generator is active.
Format:	WGO {<WaveGenID>} <StartMode>
Arguments:	<WaveGenID> is the wave generator identifier <StartMode> is the start mode for the specified wave generator. In the WGO command, you supply the start mode in hex or decimal format. More than one start mode can be set at a time as hexadecimal sum using a single WGO command. When no bits are set (<StartMode> = 0), there is no wave generator output for all axes.

	<p>Note that bit 3 cannot start the wave generator output by itself. It simply specifies a certain start option and must always be combined with one of the start modes specified in bit 0 (0x1 or 1) or bit 1 (0x2 or 2). See the examples below.</p> <p>The start mode values in detail:</p> <p>0 (decimal format), no bits are set: there is no wave generator output for the associated axis, and any Hexapod motion due to a wave generator output is stopped.</p> <p>bit 0 = 0x1 (hex format) or 1 (decimal format): start wave generator output, synchronized by servo cycle</p> <p>bit 1 = 0x2 (hex format) or 2 (decimal format): start wave generator output triggered by digital input line 1, synchronized by servo cycle.</p> <p>NOTE: the external signal must have a width of more than 1 ms. The wave generator output starts only with the first rising edge which is detected on this input line. When a trigger pulse is sent while the wave generator is running, the pulse is ignored.</p> <p>bit 3 = 0x3 (hex format) or 8 (decimal format): start trigger output on digital output line 1 (start option) Trigger output on digital output line 1 is generated as a sequence of pulses. The trigger output starts when the first point of the waveform is output by the wave generator and runs as long as a wave generator is running. The pulse width is one servo cycle, and the time between pulses is also one servo cycle, i.e. the output line toggles between HIGH and LOW synchronized with the servo cycles.</p>
Example:	Start options bit 1 and bit 3 are set simultaneously: Output of wave generators 3 is started triggered by digital input line 1. All other wave generators are started simultaneously. In addition, a sequence of pulses is output on digital output line 1 as long as the wave generator is running. WGO 3 10
Response:	None
Troubleshooting:	<p>Invalid wave generator identifier There is no wave table connected to any wave generator. Use WSL (p. 20) to connect a wave table.</p> <p>Wave generator output and move commands: When the wave generator output is active, no move command and no macro containing a move command for any axis is allowed.</p>
Notes:	<p>All waveform output is synchronized because there is a common pulse generator used by all wave generators.</p> <p>For that reason, wave tables which are supposed to run at the same time (each with one wave generator) must have the same length. Otherwise the wave generator cannot be started.</p>

WGO? (Get Wave Generator Start/Stop Mode)

Description:	<p>Gets the start modes of the given wave generator while the wave generator is active (WGO settings (p. 18)).</p> <p>Note that when the wave generator output is finished or when it was stopped by #24 or STP (see user manual of the C-887 Hexapod controller) the response to WGO? is 0.</p> <p>The #9 single-character command (p. 13) can be used to query the current activation state of the wave generators. The reply shows if a wave generator is running or not.</p> <p>For more information see “Wave Generator” (p. 3).</p>
Format:	WGO? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier
Response:	<p>{<WaveGenID>="<StartMode> LF}</p> <p>where</p> <p><StartMode> is the current start mode of the wave generator while the wave generator is active, in decimal format. The value may be the sum of several start modes. See the WGO command description for details.</p>

WSL (Set Connection Of Wave table To Wave Generator)

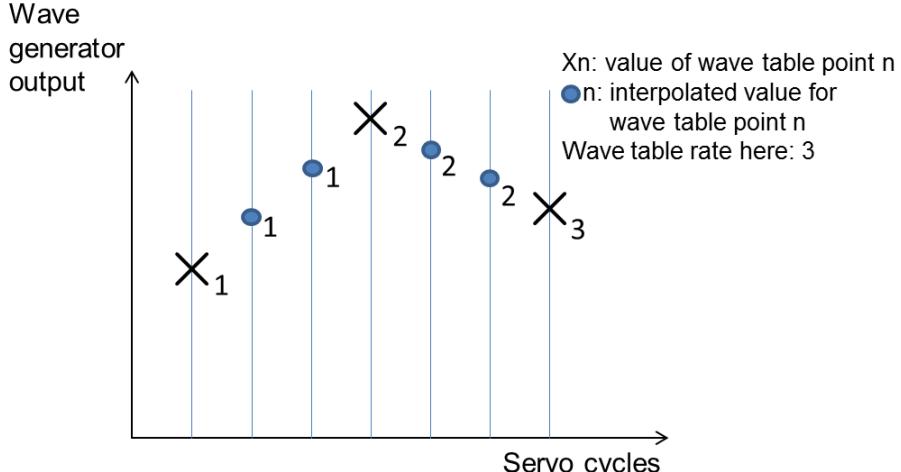
Description:	<p>Wave table selection: connects a wave table to a wave generator or disconnects the selected generator from any wave table.</p> <p>Two or more wave generators can be connected to the same wave table, but a generator cannot be connected to more than one wave table.</p> <p>As long as a wave generator is running, it is not possible to change its wave table connection.</p> <p>For more information see “Wave Generator” (p. 3).</p>
Format:	WSL {<WaveGenID> <Wave tableID>}
Arguments:	<p><WaveGenID> is the wave generator identifier. The wave generators are correlated to the axes: 1=X, 2=Y,...6=W.</p> <p><Wave tableID> is the wave table identifier. If <Wave tableID> = 0, the selected generator is disconnected from any wave table.</p>
Response:	None

WSL? (Get Connection Of Wave table To Wave Generator)

Description:	Get current wave table connection settings for the specified wave generator. For more information see "Wave Generator" (p. 3).
Format:	WSL? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier
Response:	{<WaveGenID>=<Wave tableID> LF} where <Wave tableID> is the wave table identifier. If <Wave tableID> = 0, no wave table is connected to the wave generator.

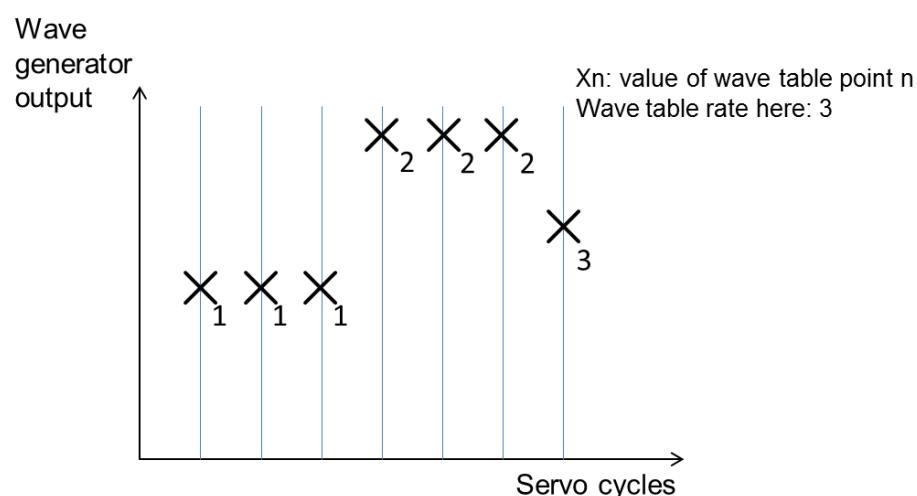
WTR (Set Wave Generator Table Rate)

Description:	Set wave generator table rate and interpolation type: Using the WTR command, you can lengthen the individual output cycles of the waveform. The duration of one output cycle for the waveform can be calculated as follows: $\text{Output Duration} = \text{Servo Cycle Time} * \text{WTR value} * \text{Number of Points}$ where Servo Cycle Time is 0.6 ms WTR value (i.e., wave table rate) gives the number of servo cycles the output of a waveform point lasts, default and minimum value is 1. Maximum value is 1000. Number of Points is the length of the waveform (i.e. the length of the wave table). Maximum value is 1000000. Thus a maximum duration of one waveform cycle equals $0.6 \text{ ms} \times 1000 \text{ WTR value} \times 1000000 \text{ data points}$. WTR sets the value of the wave generator table rate in volatile memory. The WTR value is always valid for the whole system and cannot be set separately for individual wave generators. The WTR value can be read with the WTR? command (p. 23). WTR also sets the type of interpolation to use for the wave generator output. If the WTR value is larger than 1, interpolation helps to avoid sudden position jumps of an axis controlled by the wave generator. NOTE: Interpolation is applied during the wave generator output (not inside of the wave tables). Without interpolation, the value of one wave table point will be output for the number of servo cycles determined with WTR. This may cause position jumps especially with larger output durations of the individual wave table points. With interpolation, the value of one wave table point will be adapted during the output to avoid position jumps.
--------------	--

	For more information see "Wave Generator" (p. 3). An application example can be found in "Modifying the Wave Generator Table Rate" (p. 9).
Format:	<code>WTR {<WaveGenID> <Wave tableRate> <InterpolationType>}</code>
Arguments:	<p><code><WaveGenID></code> is the wave generator identifier. When you select one wave generator using WTR all other wave generators are addressed simultaneously.</p> <p><code><Wave tableRate></code> is the table rate to be used for wave generator output (unit: number of servo-loop cycles), must be an integer value larger than zero</p> <p><code><InterpolationType></code> When a wave generator table rate higher than 1 is set, this option can be used to apply interpolation to the wave generator output between wave table points. The following interpolation types can be selected:</p> <ul style="list-style-type: none"> 0 = no interpolation 1 = straight line (default)
Examples:	<p>Interpolation type: straight line (default)</p> <p>The wave table rate of wave generator 5 is set to 3, and thus for all other wave generators, applying interpolation type 1:</p> <p><code>WTR 5 3 1</code></p>  <p>The graph shows the relationship between wave generator output (Y-axis) and servo cycles (X-axis). Vertical blue lines represent the servo cycles. Blue dots labeled '1' and '2' represent wave table points. Between these points, a straight line segment connects them. The points are labeled with their respective Xn values. A legend on the right side of the graph defines the symbols: Xn for wave table point n, n for interpolated value for wave table point n, and 'Wave table rate here: 3' indicating the rate of 3 servo loops per table point.</p>

Interpolation type: no interpolation
 The wave table rate of wave generator 5 is set to 3, and thus for all other wave generators, without interpolation:

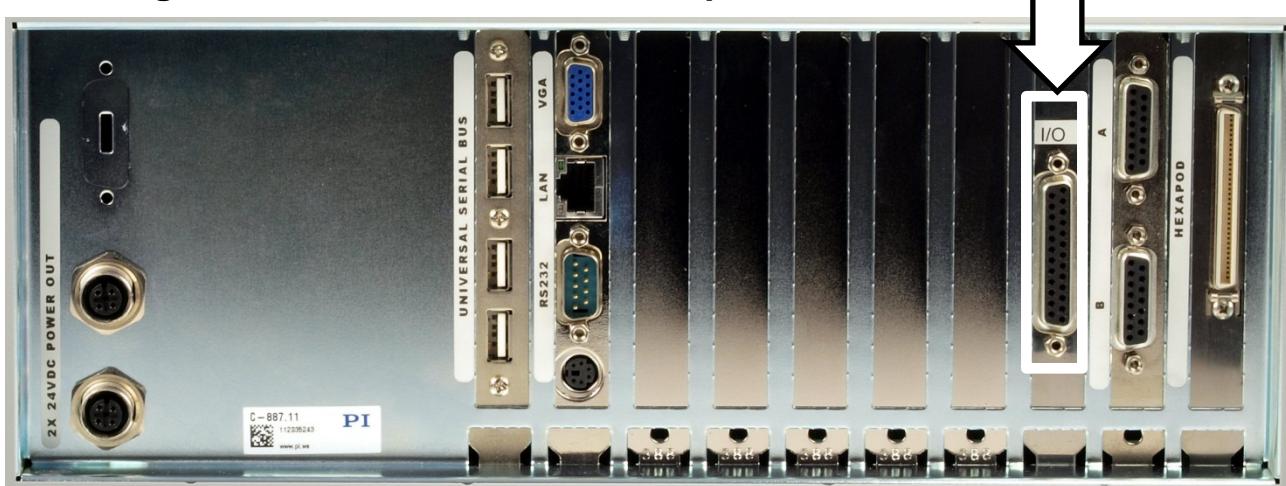
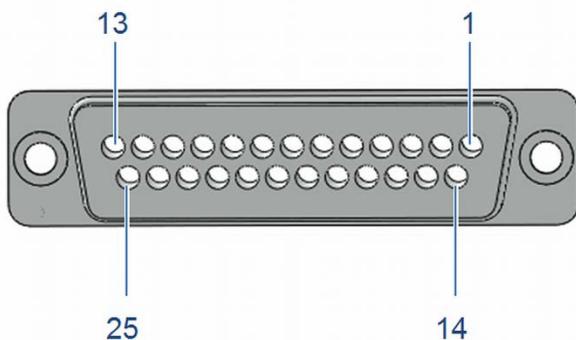
WTR 5 3 0



Response: None

WTR? (Get Wave Generator Table Rate)

Description:	Gets the current wave generator table rate, i.e. the number of servo-loop cycles used by the wave generator to output one waveform point. Gets also the interpolation type used with table rate values > 1. For more information see "Wave Generator" (p. 3). An application example can be found in "Modifying the Wave Generator Table Rate" (p. 9).
Format:	WTR? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier. Note that all wave generators use the same wave table rate.
Response:	{<WaveGenID>=<Wave tableRate> <InterpolationType> LF} where <Wave tableRate> is the table rate used for wave generator output (unit: number of servo-loop cycles) <InterpolationType> interpolation type applied to outputs between wave table points when a wave generator table rate higher than 1 is set: 0 = no interpolation 1 = straight line

Pin Assignment of I/O Interface of Hexapod Controller**Figure 1:** Rear view of the Hexapod controller**Figure 2:** Sub-D25 socket at Hexapod controller

Pin	Function
1	GND
2	Input #2
3	GND
4	Input #5
5	GND
6	Input #8
7	GND
8	Output #2
9	GND
10	Output #5
11	GND

Pin	Function
12	Output #8
13	GND
14	Input #1
15	Input #3
16	Input #4
17	Input #6
18	Input #7
19	Not connected
20	Output #1
21	Output #3
22	Output #4
23	Output #6
24	Output #7
25	Not connected

Technical Data of H811B0018 Hexapod

The H811B0018 differs from the standard Hexapod in the following technical data:

	H811B0018	Unit	Tolerance
Active axes	X, Y, Z, θ_X , θ_Y , θ_Z		
Motion and positioning			
Single-actuator design resolution	80	nm	
Min. incremental motion X, Y	1	μm	typ.
Min. incremental motion Z	0.5	μm	typ.
Min. incremental motion θ_X , θ_Y , θ_Z	7	μrad	typ.
Backlash X, Y	2	μm	typ.
Backlash Z	0.5	μm	typ.
Backlash θ_X , θ_Y	20	μrad	typ.
Backlash θ_Z	30	μrad	typ.
Max. velocity X, Y, Z	25	mm/s	
Max. velocity θ_X , θ_Y , θ_Z	625	mrad/s	
Typ. velocity X, Y, Z	10	mm/s	
Typ. velocity θ_X , θ_Y , θ_Z	250	mrad/s	
Max. actuator (strut) velocity X, Y, Z	25	mm/s	
Mechanical properties			
Load (base plate horizontal / any orientation)	3 / 1	kg	max.